UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/693,517	10/19/2000	Lawrence A. Crowl	SUN1P381/P4502	7922

24726 7590 06/06/2003

SUN MICROSYSTEMS INC
901 SAN ANTONIO RD
MS PALO1-521
PALO ALTO, CA 94303

EXAMINER
VU, TUAN A

ART UNIT	PAPER NUMBER
2124	4

DATE MAILED: 06/06/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/693,517	CROWL ET AL.	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 October 2000*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10/19/2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)	4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) ☐ Notice of Informal Patent Application (PTO-152)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .	6) ☐ Other: .

DETAILED ACTION

1. This action is responsive to the application filed October 19, 2000.

Claims 1-20 have been submitted for examination.

Claim Objections

2. Claim 12 is objected to because of the following informalities: the element "instant" recited as in 'at least one instant' and 'desired instant', lines 3-4 of claim, appears to be a misspelled case of the term "instance" and should be replaced accordingly. The examiner will treat the term as if it were "instance".

Appropriate correction is required.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Perks et al., USPN: 6,041,180(hereinafter Perks), in view of Burch, USPN: 6,308,320 (hereinafter Burch).

As per claim 1, Perks discloses a method of compilation of source program using one or more associated instances of stored class templates (e.g. *template classes* - col. 3; *comment record* – col. 6, lines 3-10), the method comprising:

identifying one or more instances available for use in the one or more class templates (e.g. col. 6, lines 3-35; col. 6, lines 45-62; *comment record* – col. 6, lines 3-10 – Note: instances

STK1.OBJ and STK2.OBJ of class templates are equivalent to identifying instances available the linking process);

receiving a first request to create a first instance during compilation of the source program (*Linker 118* – Fig. 2; col. 6, line 62 to col. 7, line 7 – Note: linker to create object file is equivalent to receive a request to create an instance of object class to add to the executable); and

determine whether the first instance has been identified in the one or more class templates (e.g. col. 5, lines 39-44; col. 6, line 65 to col. 7, line 27 – Note: checking comments and comparing CRC is equivalent to determining whether an existing instance has been created).

But Perks does not specify that the instances of class templates are libraries instances. However, Perks discloses a possibility of organizing class templates into collection classes and implementations in libraries thereof (e.g. col. 5, lines 49-57); and teaches a *comment record* to store template filename (col. 6, lines 35-62). Furthermore, Burch, in a method to compile and link compiled code into executable analogous to the linking process by Perks, discloses the use of reuse depository for storing previously created object files for reuse in subsequent linking (e.g. col. 12, lines 14-25; *reuse depository 208* - Fig. 6D). It would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the class template collection or record as suggested in the optimization method by Perks (e.g. col. 1, lines 34-46) as a repository of reusable class instances or linkable object instances, i.e. libraries, as taught by Burch because this would provide a steadier source of reusability for further compilation processes as mentioned by Burch (e.g. col. 1, lines 27-49).

As per claim 2, Perks (in combination with Burch) discloses creating the first instance when the first instance has not been identified in the libraries and creating such instance when it has been identified in such libraries (e.g. col. 1, lines 46-54; col. 5, lines 39-44 – Note: identifying matching CRC is equivalent to not recreating the class function names, or instances of library object while the step of creating a new instance is implicitly disclosed because otherwise Perks would not be able to store the instance names in the *comment records* – see col. 6, lines 8-11).

As per claim 3, Perks (in combination with Burch) discloses

identifying linker symbol names for instances available in the libraries (e.g. *Symbol table 124, comment record, template object code, template filename* -- col. 6, lines 3-11, 39-42; col. 6, lines 62 to col. 7, lines 10); and

creating the first instance when such linker symbol name is identified as not matching any available instances (e.g. col. 6, lines 3-11; Fig. 2; *comment record, CRC* - col. 6, lines 62 to col. 7, lines 10 – Note: the creating of instances of template functions, or object code identified as non-existent by Perks is implicitly disclosed for otherwise the generation of the executable would not be possible).

Although Perks does not explicitly specify that the creation of instance is operable by matching the linker symbol name for such instance with any of the identified available instances in the libraries, Perks teaches the use of symbol table (col. 6, lines 3-10) and record of template object names (col. 6, lines 39-42) to submit to the linker (Fig. 2) for identification of duplicate object code to match the CRC associated with available name instance of code, i.e. alias of names (e.g. *aliasing of function names* -- col. 5, lines 34-46; col. 6, lines 62 to col. 7, lines 10); hence, has disclosed the matching of instance names for the linker symbol names and available instance names from the libraries.

As per claim 4, Perks (combined with Burch's teachings) discloses

accessing the one or more libraries (e.g. *comment record* -- col. 6, lines 39-42);

examining linker symbol names within such libraries (col. 6, lines 62 to col. 7, lines 10 -- Note: the symbol table provided along with the comment record associate the symbol names with the name of the record of templates is equivalent to examining linker symbol names within such record).

Perks does not explicitly specify selecting symbol names that are likely to correspond to instances available in the libraries; nor saving such selected symbol names. Perks, however, discloses finding the symbol names that are required for the linking (Fig. 2) and generating address space for the instance object code identified as useful for the executable code translation process (col. 7, lines 8-49). In view of the process to identify object code instance to create and to reuse as disclosed above, the limitation as to select which symbol names to correspond to which instance of template object code stored in the libraries, or comment records (col. 6, lines 3-10, 39-42); and to save them for the address space generation as mentioned above is implicitly disclosed by Perks.

As per claim 5, Perks(in combination with Burch) implicitly discloses examining and extracting linker symbol names likely to correspond to instances of libraries code (see claim 4 for similar rejection).

As per claim 6, Perks does not disclose selecting linker symbol names that include a predetermined sequence of characters, although Perks associates with each object code name a string of CRC (col. 7, lines 14-24). Burch, in the method mentioned in claim 1, discloses associating a string of hashed characters with each object code/file name (e.g. Fig. 6D). It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide a predetermined string of characters as a hash string for naming as taught by Burch to complement the library object code identifying as mentioned by because this would add an extra dimension to the identification scheme in regard to object codes stored in libraries for a long run usage because the larger the library and number of code instances become, a more elaborate scheme for securely differentiating object code or streams, e.g. using hash value by Burch, would be more suited to handle such increase.

As per claim 7, Perks does not disclose a hash table to store linker symbol names; but in view of the teachings by Burch from claim 6 above, this limitation would have been obvious for the same rationale used therein.

As per claim 8, Perks (using Burch's teachings) discloses

obtaining a first linker symbol name for the first instance(e.g. *Symbol table 124, comment record, template object code, template filename* -- col. 6, lines 3-11, 39-42);

comparing the symbol name with those selected linker symbol names likely to correspond to template instances (e.g. col. 6, lines 62 to col. 7, lines 24 – Note: using symbol table and template object name in comment record to identifying code to load is equivalent to selecting symbol names likely to correspond to template instances), and

creating the first instance when the first linker symbol does not match any of those selected linker symbol names likely to correspond to template instances (Note: the creating of instances of template functions, or object code identified as non-existent by Perks is implicitly disclosed for otherwise the generation of the executable would not be possible).

As per claim 9, Perks discloses source program in C++ (e.g. col. 6, lines 17-29).

As per claim 10, Perks discloses a system suitable for compilation, the system comprising: a source program (e.g. *source 104*- Fig. 2); a class template record including at least one instance available for use by the program (e.g. col. 6, lines 39-62); an enhanced compiler suitable for compilation of source code (Fig. 2), such compiler accessing the class template record storage to identify the one instance available in the library (e.g. col. 6, line 65 to col. 7, line 7).

But Perks does not specify that the record storage for instances of class templates are libraries of instances. This library limitation, however, has been addressed in claim 1 above, hence is rejected herein using the same rationale set forth therein.

As per claims 11 and 12, Perks(in combination with Burch) discloses retrieving, i.e. using an instance extractor (re claim 11), an instance of the template in the record, or library, available to be use for the program generating (e.g. col. 6, line 65 to col. 7, line 7- Note: the obtaining of name from the record is equivalent to extracting such name from the record of template function objects); and comparing, i.e. using a instance name comparator(re claim 12), one instance of template available with a desired instance (e.g. *aliasing of function names, CRC* -- col. 5, lines 34-46).

As per claim 13, Perks discloses storage of an instance name (e.g. col. 6, lines 35-62).

As per claim 14, Perks discloses a method of compilation using one or more associated instances of stored class templates (e.g. *template classes* - col. 3; *comment record* – col. 6, lines 3-10), the method comprising:

examining a linker name table of one or more associated template instances (e.g. col. 6, lines 62 to col. 7, lines 10 – see Note in claim 4);

extracting from the linker name table one or more linker symbol names that are likely to correspond to the template instances (e.g. col. 6, lines 3-10, 39-42 – see corresponding rationale in claim 4);

storing the extracted symbol names (col. 7, lines 8-49 – Note: the processed symbol names destined for the creating address space and linking the target code implicitly discloses the storing the extracted names);

receiving a first request to create a first instance, said first instance having a first linker symbol name (*Linker 118* – Fig. 2; col. 6, line 62 to col. 7, line 7 – Note: linker to create object file is equivalent to receive a request to create an instance of object class to add to the executable; *symbol table* - col. 6, lines 63-66);

comparing the first linker symbol name with one or more stored linker symbol names (e.g. *aliasing of function names* -- col. 5, lines 34-46; col. 6, line 65 to col. 7, line 7 – Note: aliasing of names with an associated CRC is equivalent to comparing names of stored symbol names and a requested linker symbol name); and

creating the first instance when such symbol name is not yet stored (e.g. col. 5, lines 39-44; Note: the step of creating a new instance is implicitly disclosed otherwise the linking with a missing object code would fail).

But Perks does not specify that the storage for instances of class templates are libraries of instances. This library limitation, however, has been addressed in claim 1 above, hence is rejected herein using the same rationale set forth therein.

As per claim 15, Perks (in combination with Burch) discloses a comparing without transforming the symbol names of one or more of the libraries (e.g. col. 6, line 65 to col. 7, line 7 – Note: the CRC and the comment record content are unaffected by the matching).

As per claim 16, Perks discloses a C++ source program (e.g. col. 6, lines 17-29); but does not disclose a hash storing of symbol names. But this limitation has been addressed in claims 6 and 7 above; hence is rejected herein using the rationale set forth therein.

As per claim 17, Perks discloses a computer-readable medium (e.g. col. 7, lines 53-62) to include code for implementing a method with the limitations as recited in claim 1 and claim 2, namely the steps of identifying(instances), receiving (request), determining(instance available), creating (first instance). Hence the instant claim step limitations are herein rejected (using Perks in combination with Burch) with the corresponding rejections as set forth in claims 1 and 2 respectively.

As per claim 18, this is computer medium claim corresponding to claim 3 above, hence is rejected herein using the rejection set forth therein.

As per claim 19, this is a computer medium claim corresponding to claim 4 above, hence is rejected herein using the rejection set forth therein.

As per claim 20, this is a computer medium claim corresponding to claim 6 above, hence is rejected herein using the rejection set forth therein.

Conclusion

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pat No. 6,405,368 to Freyburger, disclosing template instantiation and dirty bit for instance status checking.

U.S. Pat No. 5,642,514 to Peckham, disclosing mapping object file to type identifier to discard duplicates.

U.S. Pat No. 5,291,601 to Sands, disclosing relocatable and shared libraries addressable via symbol table.

U.S. Pat No. 5,615,400 to Cowsar et al., disclosing catalog of registered class identifiers and lookup engine.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 746-7239, (for formal communications intended for entry)

or: (703) 746-7240 (for informal or draft communications, please label "PROPOSED" or "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

[illegible]

VAT
May 23, 2003

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100